IN THE SPECIFICATION:

Please replace paragraph [0005] with the following amended paragraph:

[0005] One problem facing programmers (or more generally any user building a query) is that databases tend to grow relatively brittle (inflexible) over time, which may increase the difficulties in crafting queries that retrieve a complete set of desired results. In other words, as business enterprise insert their own data, change data structures or formats, add features, and attempt to retire applications that once used the data or support older "legacy" applications, data may exist in the system in more than one format. For example, names may be entered in all capital letters sometimes (but not always), local area codes may be specified in some cases (~~as~~ but NULL in others), and employee IDs may be displayed in one format and stored in another format.

Please replace paragraph [0039] with the following amended paragraph:

[0039] Components of the server computer 104 may be physically arranged in a manner similar to those of the client computer 102. For example, the server computer 104 is shown generally comprising a CPU 135, a memory 133, and a storage device 134, coupled to one another by a bus 136, which may all function[[s]] as similar components described with reference to the client computer 102. The server computer 104 is generally under the control of an operating system (e.g., IBM OS/400®, UNIX, Microsoft Windows®, and the like).

Please replace paragraph [0045] with the following amended paragraph:

[0045] Referring back to FIG. 2A, the query execution component 150 is generally configured to transform the abstract query 202 into a concrete query compatible with the physical data representation (e.g., an XML query, SQL query, etc), by mapping the logical fields of the expanded abstract query ~~208~~ 202 to the corresponding physical fields of the physical data representation 214. The mapping of abstract queries to concrete queries, by the query execution component 150, is described in detail in the previously referenced co-pending application 10/083,075.

Please replace paragraph [0046] with the following amended paragraph:

[0046] As illustrated, [[The]] the abstract query 202 may include one or more query components, such as a query condition 204, that may be described by an annotation 159. The annotation 159 may be created by a first user, via the annotation system 140, during a first query building session, for example, in response to gaining a particular insight into the data targeted by the query. In a subsequent query building session, a second user may build a query with the same (or a similar) query condition 204. This query session may be monitored (e.g., by a component of the query building interface 122, or annotation system 140) to detect annotated query components. In response to detecting an annotated query component (e.g., the same query condition 204), the second user may be provided with an indication that a corresponding annotation 159 exists.

Please replace paragraph [0051] with the following amended paragraph:

[0051] At step 306, the annotation is stored with a reference to the annotated query component. For example, an annotation record 150, as illustrated in FIG. 2C may be generated and stored in the annotation store. As previously described, the reference ~~152to~~ 152 to the annotated query component may comprise any suitable information to indicate the query component. As a simple example, in some cases, the reference may be an actual code portion of a query, such as the annotated query component itself ('ICD9 = 410'). For other cases, however, the query component may be broken down into simpler components. For example, an annotated query component may comprise a group of query conditions, which may be decomposed into fragments corresponding to the individual query conditions that may be stored in the annotation record. Further, a user may be able to select and create an annotation for a particular instance value (e.g., with a common misspelling) or a field (for example, to clarify exactly what data is contained in the field).

Please replace paragraph [0055] with the following amended paragraph:

[0055] The operations 350 begin, at step 352, by monitoring one or more user-specified query components and, at step 354, annotations (if any) for the monitored query components that are retrieved. For example, as illustrated in FIG. 4C, a user may build a query to return a list of heart attack victims in their forties. During the query building process, query components (e.g., conditions, instance values, and/or fields in Conditions window 424, as well as results fields in window 426) may be monitored. For example, as new query components are added, annotation records may be searched for references to the newly added components, to determine if annotations exist for any of the specified query components. As previously described above, the types of references stored in the annotation records may vary and, therefore, the techniques used to search for annotations may vary as well.

Please replace paragraph [0056] with the following amended paragraph:

[0056] For example, rather than search for annotations using direct text searching (e.g., to find an exact match with the specified query component), a search may be performed based on a logical meaning of the query component. In other words, a specified query component may be broken down into simpler parts and common logical rules may be applied. As an example, logical rules[[,]] such as the following may be applied to find annotations for query components having the same logical meaning:

condX AND condY == condY AND condX, and

(condX AND condY) OR condZ == condZ OR (condY AND condX).

Further, references to query components that contain parameter markers, as described above, will also be searched to find annotations associated with common fields, regardless of particular instance values provided.